

Network Coding over Dynamically Changing Networks

Christina Fragouli

EPFL

Email: christina.fragouli@epfl.ch

Abstract—We here advocate the case for network coding as a guiding paradigm for the operation of networks that vary in a small time frame, due to node mobility, channel variations, and varying traffic conditions. Three ideas that appeared successively in time brought in place an elegant operation for network coding in such environments. These are, use of randomized network coding for intermediate node operation, use of generations to avoid synchronization, and use of subspace coding to allow for small packet sizes. Information theoretical performance limits support these results.

I. INTRODUCTION

Network coding is an area that has emerged during the last decade, and promises to revolutionize the way we treat information in a network by having a deep impact in all network functionalities, such as routing, network storage, and network design [1], [2]. The two most popular examples that are used to demonstrate network coding are depicted in Fig. 1 and 2. Four monographs and a book have recently been published on this subject [3], [4], [5], [6], [7] as well as a number of tutorial articles [8], [9], [10], [11]. This paper explores the application of network coding in dynamically changing networks.

By dynamically changing, we refer to networks where the structure, topology, and demands may vary in a short time scale as compared to the information transfer. For example, in a wired network, the edge capacities may vary due to changing traffic conditions and congestion. In a peer-to-peer network, thousands of nodes may join and leave the network within seconds. In a wireless network, we may have time variability due to fading channels, interference and node mobility.

The operation and management of dynamically changing networks is further challenged by the fact that these are networks where often the organization is ad-hoc, and the participating nodes have limited resources, for example in terms of communication and computational resources. Thus only low complexity, decentralized and scalable approaches can be feasibly supported.

We will attempt to make the case in this paper, that this is a situation where network coding can significantly help with, thus motivating the use of network coding in such environments. The presentation is organized as follows. In Section II we examine how network coding can be implemented in practice over dynamically changing networks. In Section III

we review information theoretical limits that characterize the performance of the discussed schemes.

II. NETWORK CODING IN A PRACTICAL NETWORK

In the two network coding examples presented in Fig. 1 and 2, we implicitly assume that there is synchronization between the network nodes, and each node performs fixed encoding operations. The receivers know these operations, and use this knowledge to decode. For example in the butterfly network in Fig. 1, x_1 and x_2 arrive simultaneously at node C. Node C always performs the same operation on these packets and forwards the resulting packet $x_1 + x_2$ to node E. The receivers R_1 and R_2 know which linear combination their received packets correspond to. For example R_1 knows it receives x_1 through edges AD and $x_1 + x_2$ through edge ED .

In a practical network, such assumptions are hard to implement. Synchronization is hard to maintain in a distributed setting. Moreover, the network structure may change quite often, for example due to varying channel conditions, nodes moving, or nodes dying. Each network change implies that we need to redesign what linear combining operations network nodes do, and accordingly inform the receivers. However, distributing information regarding the overall network structure and coding operations is costly. Thus clearly, network coding cannot be a viable solution unless it can be implemented in a decentralized manner.

Fortunately, three ideas, that appeared successively in time, give us an elegant and flexible way to perform network coding in a completely decentralized manner. These are:

- 1) Randomly chose the linear combinations at each network node [12].
- 2) Append “coding vectors” at the header of each packet to allow the receivers to decode without need of synchronization [13].
- 3) Use subspace coding to achieve the same goal more efficiently [14].

The first idea, randomized network coding, applies to the intermediate network node operation. The second and third ideas build on the use of randomized coding, and examine a complementary aspect, namely, given this mode of network operation, what coding scheme - what actions - the source and the receiver should implement. We will in fact see that the second approach can be viewed as a special case of the third approach. We proceed to discuss these ideas in more detail in the following subsections.

This work was supported by the Swiss National Science Foundation award 200021-103836/1 and by the EU project N-CRAVE, FP7 ICT-2007-215252.

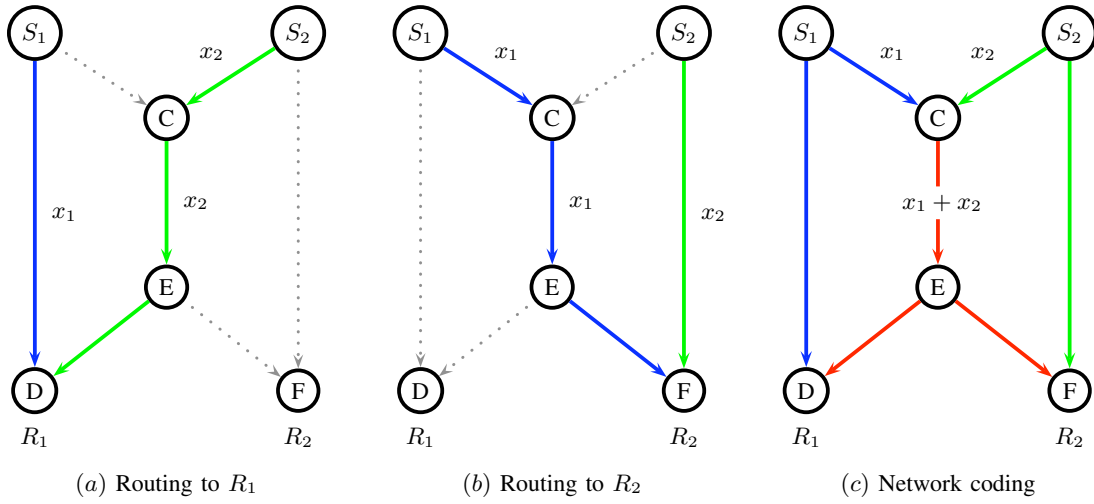


Fig. 1. The butterfly network. Sources S_1 and S_2 multicast their information to receivers R_1 and R_2 . S_1 and S_2 multicast to both R_1 and R_2 . All links have capacity 1. With network coding (by XORing the data on link CD), the achievable rates are 2 for each source, the same as if every destination were using the network for its sole use. Without network coding, the achievable rates are less (for example if both rates are equal, the maximum rate is 1.5).

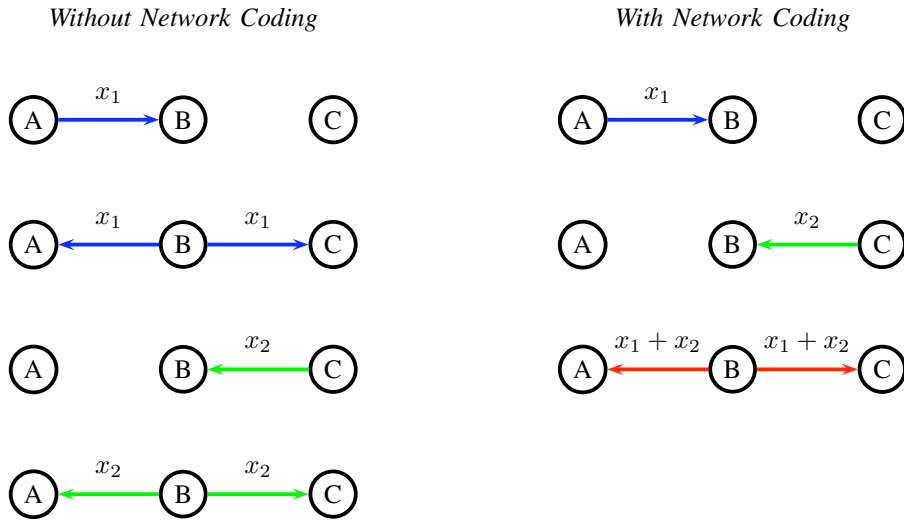


Fig. 2. Nodes A and B exchange information via relay B . The network coding approach uses one broadcast transmission less, and thus offers benefits in terms of bandwidth efficiency, delay and battery life.

A. Randomized Network Coding

Assume we have n source packets $\{x_1, \dots, x_n\}$ that contain symbols over a field \mathbb{F}_q and we want to convey them to multiple destinations over a network using network coding. Throughout the network, intermediate nodes perform linear combining of the source packets. Thus, a destination receives combinations of the form

$$c_1 x_1 + c_2 x_2 + \dots + c_n x_n,$$

where $c_i \in \mathbb{F}_q$. In the network coding literature, the vector of coefficients

$$c = [c_1, c_2, \dots, c_n]$$

is called a *coding vector*. Each destination can retrieve the data, if it receives n linearly independent combinations of the source packets, or, n linearly independent coding vectors. For example, let $\{\rho_i\}$ be the combined packets a destination

collects, we can write in a matrix form:

$$\begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_n \end{bmatrix} = \underbrace{\begin{bmatrix} c_{11} & c_{21} & \dots & c_{n1} \\ c_{12} & c_{22} & \dots & c_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{1n} & c_{2n} & \dots & c_{nn} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \quad (1)$$

If the linear combinations are independent, and matrix \mathbf{A} is full rank, we can solve the above equations and retrieve the source packets. For example, in the butterfly network in Fig. 1, the receivers need to solve systems of equations as in (1) with matrices

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

The task of network code design amounts to deciding what linear combinations to form throughout the network so that each receiver gets a full rank set of equations.

Randomized network coding is based on the simple idea that, for a field size q large enough, there exist so many valid solutions, that even random choices of the coefficients allow us to find a valid solution with high probability. Thus we can simply ask each intermediate node in the network to create and send uniform at random linear combinations of the packets it has received. The associated probability of error can be made arbitrarily small by selecting a suitably large alphabet size [15], [12]. For example, if we could choose the coefficients $\{c_{ij}\}$ of matrix \mathbf{A} in (1) uniformly at random, the matrix \mathbf{A} would be full rank with probability at least $(1 - \frac{1}{q})^n$. In practice, simulation results indicate that even for small field sizes (for example, using $m = 8$ bits per symbol, i.e., $q = 2^8$) the probability of error becomes negligible [16].

To conclude, randomized network coding requires no centralized or local information, is scalable and yields to a very simple implementation. Thus, it is very well suited to a number of practical applications, such as dynamically changing networks.

B. Generations and Coding Vectors

The next question to answer is, even if we randomly select what linear combinations to perform, how do we convey to the destinations what are the linear combinations they have received. Moreover, in a network where information gets generated at a constant rate, we need to decide what packets to combine and how often do we decode. To achieve these, we cannot rely on synchronization, since packets are subject to random delays, may get dropped, and follow different routes.

The approach in [13] first groups the packets into *generations*. Packets are combined only with other packets in the same generation. A generation number is appended to the packet headers to make this possible (one byte is sufficient for this purpose). The size of a generation can be thought of as the number of source packets n in synchronized networks: it determines the size of matrices the receivers need to invert to decode the information. Since inverting an $n \times n$ matrix requires $\mathcal{O}(n^3)$ operations, and also affects the delay, it is

desirable to keep the generation size small. On the other hand, the size of the generation affects how well packets are “mixed”, and thus it is desirable to have a fairly large generation size. Indeed, if we use a large number of small-size generations, intermediate nodes may receive packets destined to the same receivers but belonging to different generations. Characterizing this trade-off is an open research problem.

As a second step, the approach in [13] appends within *each* packet header a vector of length n that describes which linear combination of the source packets $\{x_1, \dots, x_n\}$ it contains. These vectors are what we called coding vectors. The encoded data is called the *information vector*. For example, the coding vector $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$, where the 1 is at the i th position, means that the information vector is equal to x_i (i.e., is not encoded). A packet that contains the linear combination $\rho = c_1x_1 + c_2x_2 + \dots + c_nx_n$ has the coding vector (c_1, \dots, c_n) and the information vector ρ .

The coding vectors are updated locally at each node that performs linear combining, to reflect the new linear combination of the source packets that the new packet carries. For example, if a node receives two packets with coding vectors $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$ and (c_1, \dots, c_n) , with corresponding information vectors x_i and ρ , it can create the new information vector $\alpha x_i + \rho$ for some value $\alpha \in \mathbb{F}_q$. To send this new information vector, it will use the coding vector $(c_1, \dots, c_{i-1}, c_i + \alpha, c_{i+1}, \dots, c_n)$. Combining can occur recursively and several times inside the network.

Each receiver examines the coding vectors of the packets it receives, to learn what are the linear combinations it has received. In particular, the coding vectors it receives are nothing but the rows of the matrix \mathbf{A} in (1) that determine the linear equations it needs to solve.

Appending coding vectors to packets incurs an additional overhead. For example, for a packet that contains 1400 bytes, where every byte is treated as a symbol over \mathbb{F}_{2^8} , if we have $h = 50$ sources, then the overhead is approximately $50/1400 \approx 3.6\%$.

C. Subspace Coding

The approach based on appending coding vectors is well suited for large packets where the overhead is small. In wireless networks, the situation is quite opposite: it is often the case that packets consist of a few bits. In such cases, using coding vectors can add a significant overhead.

A new approach recently proposed in [14], [17] promises to be helpful in this situation. This approach is designed to work with use of randomized network coding, and is based on using subspaces as “codewords” to convey the information from the sources to the receivers. For simplicity we will here consider a single source transmitting n independent packets to receivers, but the same approach can easily be extended to multiple sources [18].

Consider a source that would like to convey n independent source packets to receivers over a network that employs randomized network coding. Assume that each packet has length λ over \mathbb{F}_q . The n packets can take in total $M = q^{n\lambda}$

values. Thus the source, for each set of packets, has one of these values to convey.

The source can achieve this as follows. First, it selects to operate over an nL dimensional vector space V over \mathbb{F}_q , i.e., a vector space, where vectors have length nL and have elements in \mathbb{F}_q . A basis of this space consists of nL linearly independent vectors. For example, the space \mathbb{F}_2^3 has the basis

$$\{\mathbf{e}_1 = [1 \ 0 \ 0], \quad \mathbf{e}_2 = [0 \ 1 \ 0], \quad \mathbf{e}_3 = [0 \ 0 \ 1]\}.$$

A subspace π is a subset of the vector space V that is a vector space itself. We can think of subspaces as “planes” that contain the origin. For example, the space \mathbb{F}_2^3 contains 7 two-dimensional subspaces. One such subspace is $\pi_1 = \langle \mathbf{e}_1, \mathbf{e}_2 \rangle$. Another is $\pi_2 = \langle \mathbf{e}_2 + \mathbf{e}_3, \mathbf{e}_1 \rangle$. It also contains 7 one-dimensional subspaces, one corresponding to each non-zero vector. Moreover, the subspace (plane) $\pi_1 = \langle \mathbf{e}_1, \mathbf{e}_2 \rangle$ contains the three “line” (one-dimensional) sub-subspaces $\pi_3 = \langle \mathbf{e}_1 \rangle$, $\pi_4 = \langle \mathbf{e}_2 \rangle$, $\pi_5 = \langle \mathbf{e}_1 + \mathbf{e}_2 \rangle$. Therefore, we can define subspaces of lower dimension as sub-sub-spaces of higher dimensional subspaces. In the above example, we can see that $\pi_3 \subset \pi_1 \subset \mathbb{F}_2^3 = V$. We say that two subspaces are *distinct* if they differ in at least one dimension. For example, $\pi_1 = \langle \mathbf{e}_1, \mathbf{e}_2 \rangle$ and $\pi_2 = \langle \mathbf{e}_2 + \mathbf{e}_3, \mathbf{e}_1 \rangle$ are distinct.

The source selects a codebook of M distinct subspaces, and each set of n packet is mapped to a different such subspace. The receivers learn this codebook. To convey the value of the source packets, the source needs to convey what is the particular subspace these packets are mapped to. To do so, it inserts in the network a set of basis vectors (packets) that span the subspace. Assume for example it sends the vectors $\{b_1, \dots, b_k\}$ that span a subspace π . The critical observation is that, the mixing through randomized network coding intermediate nodes perform, preserves the subspaces. Indeed, linear operations, no matter what these operations are, can only create vectors that are in the span of the basis $\{b_1, \dots, b_k\}$ and thus within π . As a result, every node that receives k linearly independent vectors will be able to identify which is the subspace π that the source has sent. The source has then transmitted information through the choice of the subspace that it sends. This property makes the use of subspaces for encoding robust to the topology of the network and to arbitrary linear operations performed at the intermediate nodes.

Using coding vectors is a special case of subspace coding [14]. We can see this through an example. Assume the source has $n = 2$ packets of length $\lambda = 2$ bits each. Assume that the first packets is $x_1 = [x_{11} \ x_{12}]$ and the second packet is $x_2 = [x_{21} \ x_{22}]$. Using the coding vector approach, the source sends one packet consisting of the coding vector $[1 \ 0]$ followed by the information vector $[x_{11} \ x_{12}]$ and another packet consisting of the coding vector $[0 \ 1]$ followed by the information vector $[x_{21} \ x_{22}]$. Let

$$b_1 = [1 \ 0 \ x_{11} \ x_{12}], \quad b_2 = [0 \ 1 \ x_{21} \ x_{22}].$$

We can think of these two packets that the source sends as spanning a 2-dimensional subspace of the 4-dimensional space \mathbb{F}_2^4 . Each time the source has a new set of packets, it will send

a different such subspace; in total the source will send one out of 16 distinct subspaces, since it observes 16 different values.

This corresponds to a particular choice of subspaces, in the subspace coding scheme. Observe that \mathbb{F}_2^4 contains not only 16, but in fact 35 distinct 2-dimensional subspaces. Thus, the sources, using the same packet length, could have conveyed a much higher rate to the receivers, by incorporating in the codebook all the 2-dimensional subspaces available. Alternatively, using the subspace approach, we can convey the same information with smaller packet length, and dispense from the coding vector overhead. This promising approach has just started to be explored in the literature.

III. THEORETICAL PERFORMANCE LIMITS

We here present information theoretical bounds for the schemes we discussed in the previous section.

A. The main theorem in network coding

For some traffic scenarios, network coding effectively allows the nodes of the network to achieve the optimal performance while operating in a decentralized fashion. This property is directly implied by the information theoretical proof of the main theorem in network coding [1], [2]. The theorem can be formally stated as follows.

Theorem 1. *Consider a directed acyclic graph $G = (V, E)$ with unit capacity edges, h unit rate sources located on the same vertex of the graph and N receivers. Assume that the value of the min-cut to each receiver is h . Then there exists a multicast transmission scheme over a large enough finite field \mathbb{F}_q , in which intermediate network nodes linearly combine their incoming information symbols over \mathbb{F}_q , that delivers the information from the sources simultaneously to each receiver at a rate equal to h .*

The information theoretic proof of the main theorem in [1] shows that if all nodes in the network do *exactly the same operation*, randomly combine their incoming flows and transmit them to their outgoing edges, no matter what is their position in the network and what the network topology between the source and the destination is, we can achieve the min-cut rate.

In other words, *even if we have a random network* between the source and the destination, and we know nothing of its structure, provided the min-cut to the receiver is maintained, and allowing all nodes to operate in exactly the same fashion, allows the receiver to get information at the min-cut rate. This is not possible in the case of routing, where information would be routed differently, depending on the network structure.

Additionally, all nodes in the network with min-cut equal to h , would inherently receive information at this rate. That is, we do not need to differentiate the network operation, depending on whether we have one or multiple receivers, provided that the transmitted rate is smaller or equal to the smallest min-cut of each intended receiver.

In some situations, we may be interested in the rate at which network nodes receive new information. The following

theorem considers the rate at which a node observes innovative information.

Theorem 2. *Consider a network operation that employs randomized network coding over a field \mathbb{F}_q . Then each node i receives innovative packets from the set of source S at a rate that is upper-bounded by $\min\text{-cut}(S, i)$. It receives innovative packets at a rate exactly equal to $\min\text{-cut}(S, i)$, if the network is in “steady state” (all network edges are used) and the field size q is sufficiently large.*

The proof uses the algebraic approach proposed by Koetter and Medard [19] to express the transfer matrix between each node i and the source, and the Schwartz-Zippel lemma to upper bound the probability that randomly chosen values for the linear combinations lead to a transfer matrix with rank equal to $\min\text{-cut}(S, i)$. The detailed proof is given in [20].

B. Noncoherent communication

We here start by presenting a formal model for subspace coding, and then use this model to derive information theoretic performance bounds.

Consider a network where nodes perform uniform at random network coding over a finite field \mathbb{F}_q . We discuss first the case of a single source and a single receiver. We assume slotted time, and a “block” time-varying channel. At timeslot l , the receiver observes

$$Y(l) = G(l)X(l), \quad (2)$$

where $X(l)$ is an $m \times T$, $G(l)$ is an $n \times m$ and $Y(l)$ is an $n \times T$ matrix defined over the finite field \mathbb{F}_q (in the rest of the paper we will omit for convenience the index l). That is, at each time slot, the receiver receives n packets of length T , that depend on a set of m packets of length T sent by the source. The source packets are independent from time-slot to time-slot. This block operation of the channel, where the received packets Y depend on a different set of sent packets X , is exactly like the standard network coding model in [1].

The block length T can be interpreted as the coherence time of the channel, during which the transfer matrix G remains constant. T is finite and fixed. If T were arbitrarily large, we could send a set of “training symbols” of finite length for the receiver to learn G and then communicate using perfect channel knowledge. In fact, this approach exactly corresponds to the use of generations we discussed earlier. Information packets are divided in generations, the number m of source packets corresponds to the generation size, and the training symbols are the coding vectors appended to the information packets. This approach leads to a rate loss that becomes pronounced as T decreases.

In our model, the transfer matrix G changes independently from timeslot to timeslot, according to the uniform at random linear combining performed by the network intermediate nodes. Although in general matrix G has some structure related to the topology of the network (see for example [21]), we will here assume that the entries of G are selected according

to the uniform distribution. We argue that this is a reasonable choice, especially for large scale dynamically changing networks, because: (i) in large networks with high probability all the elements of matrix G will be random variables (no constant elements), and (ii) the network topology changes introduce additional randomness in the matrix structure. The model given in (2) along with the modeling for G given above is clearly information stable and hence the capacity is given by

$$C = \sup_{p(x)} \frac{1}{T} I(X; Y),$$

where $p(x)$ is the input distribution. For a coding strategy that induces an input distribution $p(x)$, the achievable rate is

$$R = \frac{1}{T} I(X; Y).$$

The generalization of this model to multiple receivers is straightforward. We thus next consider the case of multiple sources, and the multiple access channel corresponding to (2). This can be expressed as

$$Y(l) = \sum_{u=1}^N G_u(l)X_u(l) = G_{MAC}(l)X_{MAC}(l), \quad (3)$$

where we have N sources, each source u inserting m_u packets in the network. Thus $X_u(l)$ is an $m_u \times T$, $G_u(l)$ is an $n \times m_u$ and $Y(l)$ is an $n \times T$ matrix over \mathbb{F}_q . We can also collect all $G_u(l)$ in the $n \times \sum_{u=1}^N m_u$ matrix G_{MAC} and all $X_u(l)$ in the $\sum_{u=1}^N m_u \times T$ matrix $X_{MAC}(l)$. Each source u then controls m_u rows of the matrix $X_{MAC}(l)$.

The models (2) and (3) can easily be extended to include noise. For example, introducing erasures in (2) can be modeled by randomly removing rows of matrix Y , or removing rows of matrix X . These operations correspond to making all zero a row or a column of matrix G . Additive noise can be introduced through a matrix $Z(l)$

$$Y(l) = G(l)X(l) + Z(l), \quad (4)$$

that follows a given distribution. Constraining the rank of matrix $Z(l)$, for example to be smaller or equal to k , corresponds to the error constraints of the channel model in [14]. In the rest of this section we will focus our attention to the noiseless case, given in (2) and (3), and derive upper and lower bounds.

Single Source

Consider the input distribution where all subspaces of the same dimension are chosen with the same probability. That is,

$$\Pr(\langle X \rangle = \pi, \dim(\pi) = r) = \alpha_r \left[\begin{matrix} T \\ r \end{matrix} \right]_q^{-1}, \quad (5)$$

where $\sum_{r=0}^{\min(m, T)} \alpha_r = 1$. The following theorem holds [22].

Theorem 3. *For large finite field size q , the rates up to*

$$\frac{1}{T} \times [\Delta(T - \Delta) + \mathcal{O}(q^{-1})] \log_2 q$$

are achievable, where $\Delta = \min(m, n, \lfloor T/2 \rfloor)$.

For the special case where $n = 1$, we get that

$$I_{\max}(X; Y) = (T - k) \log_2 q - q^{-k} \log_2 k - (1 - q^{-k}) \log_2 \left(\frac{1 - q^{-k}}{1 - q^{-T}} \right).$$

For large q the mutual information behaves as

$$I(X; Y) \approx T - k, \quad (6)$$

which is maximized for $k = 1$. Thus, selecting matrices X that span one-dimensional subspaces leads to an achievable rate of $1 - 1/T$ which is close to the trivial upper bound of 1.

The proof of the following theorem is provided in [20].

Theorem 4. *If $T > n + \min(m, n)$, there exist a number q_0 such that for $q > q_0$, the optimal input distribution is as in (5) with $\alpha_{\min(m, n)} = 1$. Then,*

$$C = [\min(m, n)(T - \min(m, n)) + \mathcal{O}(q^{-1})] \log_2 q.$$

It is worth noting that the above equation coincides with the lower bound in Theorem 3, since under the assumptions of Theorem 4 we have that $\min(m, n, \lfloor T/2 \rfloor) = \min(m, n)$.

Multiple Sources

For simplicity we consider the case of two sources X_1 and X_2 . The well known rate region for the MAC channel is given by the union of rate pairs satisfying [23]

$$R_1 \leq I(X_1; Y|X_2) = H(Y|X_2) - H(Y|X_1, X_2),$$

$$R_2 \leq I(X_2; Y|X_1) = H(Y|X_1) - H(Y|X_1, X_2),$$

$$R_1 + R_2 \leq I(X_1, X_2; Y) = H(Y) - H(Y|X_1, X_2),$$

for a given channel probability $\Pr(Y|X_1, X_2)$ and $P(X_1, X_2) = \Pr(X_1)\Pr(X_2)$. Consider the input distribution

$$\Pr(\langle X_i \rangle = \pi_i, \dim(\pi_i) = r_i) = \begin{cases} \left[\begin{matrix} T \\ k_i \end{matrix} \right]_q^{-1} & r_i = k_i, \\ 0 & \text{otherwise,} \end{cases}$$

for $i \in \{1, 2\}$ where k_i are some fixed values satisfying $0 \leq k_i \leq \min(m_i, T)$. Substituting this distribution in the entropy expressions we have previously calculated, we get the achievable region

$$R_i \leq \frac{1}{T} [T \min(n, k_i) + k \max(n - k_i, 0) - n \min(k, T)],$$

$$R_1 + R_2 \leq \frac{1}{T} [[T - \min(n, k, T)] \min(n, k, T)] + \frac{1}{T} [n [\min(n, k, T) - \min(k, T)]],$$

where $k = k_1 + k_2$, $0 \leq k_i \leq \min(m_i, T)$, and $i \in \{1, 2\}$. The complete proof is provided in [20].

Maximizing the above equations over different values of k_1 and k_2 , we obtain the following theorem.

Theorem 5. *For the case of two sources, an achievable region is given by*

$$R_i \leq \frac{1}{T} [\Delta_i (T - \Delta_i)],$$

$$R_1 + R_2 \leq \frac{1}{T} [\Delta (T - \Delta)],$$

where $m = m_1 + m_2$, $\Delta_i = \min(m_i, n, \lfloor T/2 \rfloor)$, $\Delta = \min(m, n, \lfloor T/2 \rfloor)$, and $i \in \{1, 2\}$.

IV. CONCLUSIONS

In this paper we argued that for some traffic scenarios, network coding effectively allows the nodes of the network to achieve the optimal performance while operating in a randomized decentralized fashion. This finds immediate application in dynamically changing environments, where centralized network management and control has a prohibitive complexity, and thus network nodes need to operate in a distributed fashion without using knowledge of the overall network configuration. We presented an approach that allows to make these ideas have impact in practice, and discussed associated information theoretical bounds.

REFERENCES

- [1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. on Information Theory*, pp. 1204–1216, July 2000.
- [2] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. on Information Theory*, vol. 49, pp. 371–381, Feb 2003.
- [3] R. W. Yeung and N. Cai, "Network error correction, i: basic concepts and upper bounds," *Commun. Inf. Syst.*, vol. 6, pp. 19–35, 2006.
- [4] N. Cai and R. W. Yeung, "Network error correction, ii: lower bounds," *Commun. Inf. Syst.*, vol. 6, pp. 37–54, 2006.
- [5] C. Fragouli and E. Soljanin, "Network coding: Fundamentals," *Foundations and Trends in Networking*, vol. 2, pp. 1–133, 2007.
- [6] —, "Network coding: Applications," *Foundations and Trends in Networking*, vol. 2, pp. 135–269, 2008.
- [7] T. Ho and D. S. Lun, "Network coding: An introduction," *Cambridge University Press, Cambridge, U.K.*, 2008.
- [8] C. Fragouli, J. Widmer, and J. LeBoudec, "Network coding: An instant primer," *ACM SIGCOMM Computer Communication Review*, 2006.
- [9] M. Effros, R. Koetter, and M. Medard, "Breaking network lojams," *Scientific American*, 2007.
- [10] P. Chow and Y. Wu, "Network coding for the internet and wireless networks," *Micorsoft Research MSR-TR-2007-70*, 2007.
- [11] "Network coding: networking's next revolution?" *Network Word*, 2007.
- [12] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, pp. 4413–4430, Oct. 2006.
- [13] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Allerton*, Oct. 2003.
- [14] R. Koetter and F. Kschischang, "Coding for errors and rrasures in random network coding," *ISIT*, June 2007.
- [15] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," vol. 27, pp. 701–717, 1980.
- [16] Y. Wu, P. A. Chou, and K. Jain, "A comparison of network coding and tree packing," *ISIT 2004*, 2004.
- [17] D. Silva and F. R. Kschischang, "Using rank-metric codes for error correction in random network coding," *ISIT*, June 2007.
- [18] M. JadariSiavoshani, C. Fragouli, and S. Diggavi, "Non-coherent network coding for multiple sources," *ISIT*, 2008.
- [19] R. Koetter and M. Medard, "Beyond routing: an algebraic approach to network coding," *INFOCOM*, vol. 1, pp. 122–130, June 2002.
- [20] M. J. Siavoshani, C. Fragouli, and S. Diggavi, "Noncoherent multisource network coding," *EPFL Technical Report*, 2008.
- [21] M. Jafarisiavoshani, C. Fragouli, and S. Diggavi, "On subspace properties for randomized network coding," *Information Theory Workshop (ITW)*, July 2007.
- [22] M. J. Siavoshani, C. Fragouli, and S. Diggavi, "Noncoherent multisource network coding," *ISIT*, 2008.
- [23] T. Cover and J. Thomas, "Elements of information theory," *Wiley Series*, 2006.